PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

register transfer diagram and declaration SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **register** **transfer** **diagram** and **declaration**

Found **17,498** of **132,857**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200

Best 200 shown

Result page: **1** 2 3 4 5 6 7 8 9 10 next

Relevance scale □□□■■

1 Diagrammatic function description of microprocessor and data-flow processor

Gotaro Odawara, Masahiro Tomita, Ichiro Ogata

June 1985 **Proceedings of the 22nd ACM/IEEE conference on Design automation**

Full text available: pdf(416.19 KB) Additional Information: full citation, abstract, references, index terms

This paper discusses a description technique for various kinds of processors. This technique is based on the Symbolic Functional Description Language, which allows logic designers to describe the behavior of hardwares at the register-transfer level in the top-down approach. The SFDL has been applied to the description of the internal behavior of a microprocessor and a data-flow processor. As a result, the SFDL has made it possible to describe the instruction set and the behavior ...

2 A symbolic functional description language

Gotaro Odawara, Jun Sato, Masahiro Tomita

June 1984 **21st Proceedings of the Design Automation Conference on Design automation**

Full text available: pdf(638.56 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a new diagrammatic hardware description language SFDL (Symbolic Functional Description Language) and a hierarchical logic design supporting system LDSS (Logic Design Supporting System). SFDL has three features that help designers design logic circuits easily and speedily; easy to describe with its simple rule, comprehensible to grasp the behavior of the circuit and suitable for computer processing. Besides, the LDSS allows designers to draw diagrams without t ...

3 N.mPc: A retrospective

Charles W. Rose, Greg M. Ordy, Frederic I. Parke

June 1983 **Proceedings of the twentieth design automation conference on Design automation**

Full text available: pdf(705.67 KB) Additional Information: full citation, abstract, references, citings, index terms

N.mPc, a mature, UNIX*-based computer-aided design tool is described. Its structure, performance, and limitations are discussed together with its applications to the design of multiple processor hardware/software systems and VLSI.

4 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

5 Microcode development for microprogrammed processors
J. P-C Hwang, C. A. Papachristou, D. D. Cornett
December 1985 **ACM SIGMICRO Newsletter , Proceedings of the 18th annual workshop on Microprogramming**, Volume 16 Issue 4
Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings, index terms

The aim of this paper is to develop a top-down design automation tool for digital system design such as microprogrammed processors. The package contains a hardware description language to specify the design, a microcode development module to generate an efficient micropogam for the microprogrammed processor's control, and a functional simulator module to verify the validity of the design. The goal of this project is to develop an interactive computer-aided design environment for specificat ...

6 A language for the description of digital computer processors
John A. Darringer
July 1968 **Proceedings of the fifth annual 1968 design automation workshop on Design automation**
Full text available: pdf(930.58 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper discusses the need for an algorithmic language for writing behavioral descriptions of digital computer processors and proposes an Algol-like candidate. It also discusses programs that simulate and automatically implement processors described in the language.

7 MIDL - a microinstruction description language
Marleen Sint
December 1981 **Proceedings of the 14th annual workshop on Microprogramming**
Full text available: pdf(848.63 KB) Additional Information: full citation, abstract, references, citings, index terms

A microinstruction description language called MIDL is introduced. A MIDL description of a microarchitecture defines the semantics and triggering conditions of all microoperations. It also defines operand selection. MIDL incorporates a timing model that allows detailed specification of the timing of each microoperation, and a sequencing model that allows the description of many different sequencing schemes.

8 Agent communication transfer protocol
Alexander Artikis, Jeremy Pitt, Christos Stergiou
June 2000 **Proceedings of the fourth international conference on Autonomous agents**
Full text available: pdf(990.14 KB) Additional Information: full citation, references, index terms

Keywords: agent communication protocols, agent conversations, agent-oriented middleware, communication models, multi-agent systems, software agents, transfer protocols

9 A study of the application of compiler techniques to the generation of micro-code
A. Keith Tirrell
May 1973 **Proceedings of the meeting on SIGPLAN/SIGMICRO interface**

Full text available: pdf(913.64 KB) Additional Information: full citation, abstract, citings, index terms

With the advent of writable control memories, the opportunity for using "firmware" to implement software systems increases dramatically. This paper presents one approach to the problem of constructing a micro-program compiler which uses a machine-independent language. While this kind of compiler language may not generate micro-programs as efficient as those constructed by assembly language, the improvement in documentation, the ease of locating logic errors, and the p ...

10 Computing curricula 2001
September 2001 **Journal on Educational Resources in Computing (JERIC)**

Full text available: pdf(613.63 KB) html(2.78 KB) Additional Information: full citation, references, citings, index terms

11 A digital system modeling philosophy and design language
Mehmet B. Baray, Stephen Y. H. Su
June 1971 **Proceedings of the June 1971 design automation workshop on Design automation**

Full text available: pdf(1.13 MB) Additional Information: full citation, abstract, references, citings, index terms

With the increasing complexity of digital systems, a new approach to system decomposition and modelling must be developed. Such an approach must not only permit a system to be described in a modular fashion, but the basic system modules must be able to be decomposed to allow a precise description of the module function. In this paper, a modelling philosophy and language are presented which permit the system designer to: (1) specify the system design in a hierarchical, modular fashion; (2) d ...

12 An interactive logic activity modelling program
H. B. Rigas, H. B. Jacoby, V. B. Hunt
August 1972 **Proceedings of the ACM annual conference - Volume 2**

Full text available: pdf(521.48 KB) Additional Information: full citation, abstract, references, index terms

Simulation of logic networks can be a valuable tool in such areas as teaching the principles of logic design, investigating hardware implementation of arithmetic algorithms, and verifying the validity of test sequences in fault diagnosis of various types of networks to mention only a few possible applications. To be useful for a large number of problems in these classes, the simulator should be both flexible in its application and convenient to use. A system is described here which allows a ...

Keywords: Graphical input simulator, Hardware simulator, Interactive logic simulator

13 High-level synthesis: technology transfer to industry
Robin C. Sarma, Mark D. Dooley, N. Craig Newman, Graham Hetherington
January 1991 **Conference proceedings on 27th ACM/IEEE design automation conference**

Full text available: pdf(677.92 KB) Additional Information: full citation, abstract, references, citings, index terms

High-level synthesis has been an active research area for ten years. However, the progress in research has yet to find its way into usage in industry. This paper describes the technology transfer of high-level synthesis at Texas Instruments. It explains how we have extended one research project, the System Architect's Workbench from Carnegie-Mellon University, into a tool that designers can use. Significant enhancements were made to the Workbench to enable it to make architectural tradeoffs ...

14 An ALGOL-like computer design language
Yaohan Chu
October 1965 **Communications of the ACM**, Volume 8 Issue 10

Full text available: pdf(1.39 MB) Additional Information: full citation, references, citings, index terms

15 A meta-compiler as a design automation tool
R. Mandell, G. Estrin
January 1966 **Proceedings of the SHARE design automation project**

Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, citings, index terms

In the field of design automation, in general and in the automatic design of digital computers in particular, it is frequently necessary to translate machine descriptions and designer requests from the input language to the language of the design automation system. This process is very similar to the translation of artificial computer programming languages into machine language. Hence, it is reasonable to attempt to apply the techniques developed for writing compilers to the task of transla ...

16 Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science
William F. Atchison, Samuel D. Conte, John W. Hamblen, Thomas E. Hull, Thomas A. Keenan, William B. Kehl, Edward J. McCluskey, Silvio O. Navarro, Werner C. Rheinboldt, Earl J. Schweppe, William Viavant, David M. Young
March 1968 **Communications of the ACM**, Volume 11 Issue 3

Full text available: pdf(6.63 MB) Additional Information: full citation, references, citings

Keywords: computer science academic programs, computer science bibliographies, computer science courses, computer science curriculum, computer science education, computer science graduate programs, computer science undergraduate programs

17 Compiler transformations for high-performance computing
David F. Bacon, Susan L. Graham, Oliver J. Sharp
December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms, review

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis of scalar quantities and data-flow techniques. In contrast, optimizations for high-performance superscalar, vector, and parallel processors maximize parallelism and memory locality with transformations that rely on tracking the properties o ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimization, parallelism, superscalar processors, vectorization

18 An undergraduate compiler laboratory
Frank Friedman, Judith A. Stebulis
January 1979 **ACM SIGCSE Bulletin , Proceedings of the tenth SIGCSE technical symposium on Computer science education**, Volume 11 Issue 1
Full text available: pdf(648.06 KB) Additional Information: full citation, abstract, references, citings, index terms

A one semester, upper-division undergraduate course in compiler techniques is described. The course is based upon the material contained in Chapter 5 of the text Algorithms + Data Structures &equil; Programs, by Niklaus Wirth. The goals of the course are (1) to introduce students to the fundamental concepts of the design and translation of higher level languages, and (2) to provide an introductory exposure to the related l ...

19 Production logic synthesis
John Darringer, Daniel Brand, William H. Joyner, Louise Trevillyan, John V. Gerbi
March 1985 **Proceedings of the 1985 ACM thirteenth annual conference on Computer Science**
Full text available: pdf(396.76 KB) Additional Information: full citation, references, index terms

20 Simulating modular microcomputers
Frank J. Langley, Gerald A. LaGro, Joan Sheehan
March 1978 **Proceedings of the eleventh annual simulation symposium**
Full text available: pdf(1.47 MB) Additional Information: full citation, abstract, references, citings, index terms

The commitment of microprocessor-based system configurations to detailed logic design and breadboard fabrication traditionally results in a costly development cycle. This paper reports on the use of a computer design high-order-language (HOL) to simulate micro-computer functional elements, "macromodules", at the register level, and verify the timing and interface requirements for a family of microcomputer configurations. The definitions of these microcomputer macromodules (i. e. ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next